

## 1. Task Name: Conduct Critical Design Review

### 2. Purpose:

To propose and explain the detailed technical design and provide all software engineering groups an opportunity to review and comment on the proposed technical solutions including finalizing all items requiring change.

### 3. Roles:

Technical designer presents the design proposal to the functional analysts, database engineers, developers, systems software engineers, system testers, and customer support personnel for review and comments.

### 4. Entrance Criteria:

- a. Updated Technical Design Schedule (S-PM-011)
- b. Documented Risks (S-PM-013)
- c. Completed Preliminary Design Peer Review Report (S-SE-001)
- d. Review Report Standard (S-SE-001)
- e. Review Defect Report Standard (S-SE-002)
- f. Updated Technical Design section of SCR (S-CM-002)
- g. Review Checklist Standard (S-PM-018)

## 5. CDR Procedures

### *Conduct Critical Design Review*

#### Purpose

The Critical Design Review (CDR) is conducted by the software developer to verify that the modified detailed system design is complete, correct, satisfies both functional and technical system requirements, and adheres to standards as identified in the SQA Plan. During the CDR, design documents (e.g., modified detailed design specifications, interface specifications and program specifications, when applicable) are evaluated to ensure that all of the information for program change development is present and to establish the integrity of the new design prior to coding and testing. The CDR may be held in increments to discuss one or more SCRs. Informal reviews may or may not include a formal meeting, but are subject to the same reporting requirements as described below for formal reviews. Design documents are also reviewed for design standards. An SQA process review of the CDR is recommended. See the Periodic Processes Phase for procedures for this review. The majority of the work performed by this task is the responsibility of the FSA.

## **Process**

### **5.1 Schedule Critical Design Review**

The individual project teams will schedule a CDR when the application has completed the detail level table design and the high-level module design. The detail level table design is defined as follows:

- All tables are mapped
- All columns are mapped

The high-level module design is defined as follows:

- each module is identified to meet the requirements,
- all table usage's and column usage's are mapped,
- all module types are identified and defined (i.e., the module network is defined),
- arguments/parameters are identified and defined,
- menus are designed, and
- roles are defined and assigned.

### **5.2 Conduct the Review**

The SQA team will conduct the review with the individual project team. The project team will be responsible for providing all information to the SQA team to perform the review. The project teams will need to perform the following activities in advance in order to provide the necessary information to the SQA team to conduct the review.

The SQA team will assist the project teams in preparing for the CDR as necessary.

The following sections describe the steps the project teams will need to perform in preparation for the CDR. They list reports that should be generated from Oracle Designer. They also list the type of information that the project teams will get from each of the reports.

#### **5.2.1. Generates the appropriate quality assurance reports for the integrated requirements from Oracle Designer for each application.**

These reports show information that will cause errors in the generation process and should be excluded from the design. These problems should be documented in the review checklists in Section 5.3

<b>Report</b>	<b>Report Purpose</b>	<b>Report Shows</b>
Invalid Database Objects Quality Control	Highlights any problems with database objects.	1. Database objects whose names are reserved words in PL/SQL or Oracle Designer 2. Oracle database objects which are defined on databases other than an Oracle database
Complete Status Quality Control	Highlights any problems with the definition completeness of database objects.	1. Database objects defined as complete but which are defined on objects that are not defined as complete (not ready to be generated by the Server Generator) 2. Database objects defined as not

complete  
 3. Database objects defined as complete but which have not yet been granted to any database users or group of users

### 5.2.2. Review deliverables by application, for completeness, accuracy, maintainability, and reliability. in accordance with established DFAS and Oracle guidelines.

These reports show details of the design. The project team should use these reports to fill out the review checklists found in section 5.3.

<b>Report</b>	<b>Report Purpose</b>	<b>Report Shows</b>
Table definition	Details of tables, views and snapshots. The information includes descriptions, volumes, column details and indexes.	Tables, views, and snapshots and their User Help Text, Volumes (start and end row numbers), Indexes, Primary Keys, and Foreign Keys. Shows columns and there User Help Text.
PL/SQL Module Definition	Depicts the definition of each PL/SQL procedure defined in the repository.	PL/SQL modules and their purpose, whether the module has been marked complete, whether the modules can correctly read and write to the database.
Column Definition	Comprehensive column definition details for the given table(s), view(s) or snapshot(s), together with the display parameters.	Columns and their Hint Text, Help Text, and Default values. Can also be used to review if the columns are in the appropriate order in each table.
Constraint Definitions	Details of the constraints defined for a given table, view or snapshot (i.e., primary key constraints, unique key constraints, foreign key constraints and check constraints.)	Tables and their Primary Key columns and the constraints, Foreign Key columns, constraints, and the tables they reference.
Database Trigger	Details of the database triggers defined for each table.	Database triggers and their purpose, and whether the trigger has been marked complete.
Database Synonym Defini	Details of the database synonyms defined for objects in the specified application system.	Synonym is ready for generation.
Columns in a Domain	Lists the tables and columns that exist in each domain.	The table and column names and the associated domain detail
Tables, Columns and Foreign Key Derivations	Lists all the columns for each table, and provides foreign key details where applicable.	Review foreign key constraints
Sequence Definition	Details of the sequence definitions.	Review sequence definitions
Modules in an Application System	Summary of all the modules. The information includes the filename, task, estimate and purpose for each module.	Status of each module (whether completed, started, or not started)
Database Table and Index Size Estimates	Estimates of space required to store the table and index database objects defined. The report also estimates the size of the indexes created implicitly, by the primary key / unique key definition on the tables.	Capacity requirements of the database to be generated for both tables and indices. The report also shows formulae used for the calculation, so the designers can

The names of these indexes are assumed to be the same as the constraint names.

Tables and indexes or constraints are listed by tablespace within the database. If the table, index or constraint has not been assigned to a tablespace, it will be listed under a tablespace named UNSPECIFIED.

The report estimates the total database size as the combined size of the tables and indexes. Table size and index size estimates are done separately. The formulae used for estimating table and index sizes are given at the beginning of the report.

make modifications to formulae as desired.

Finally, report shows quality control errors that would not allow the report to be generated.

**Note that this report will not generate any data unless the tables... are mapped to tablespaces. The undefined tablespace does not work.**

### 5.2.3. Review requirements traceability reports that compare tables to entities and modules to functions.

A major factor in the success of the DFAS design is the ability to trace tables back to entities and modules (both server side and client side modules) back to functions. The following reports will show this tracing. The designers should use these reports to fill out the review checklists in section 5.3.

Report	Report Purpose	Report Shows
Entity to Table Implementation	List of the entities and the tables that implement the entities, and a list of the tables and the entities that each table implements.	Tables that have not been generated from entities, and columns that have not be generated from entities. Any of these should have Design Comments describing why they have been created
Module Definition	Full details of the modules that implement functions.	Modules that have not been generated from Functions. <b>This report seems to be messed up, prints funny.</b>
PL/SQL Module Definition	Depicts the definition of each PL/SQL procedure defined in the repository.	PL/SQL modules that have not been generated from functions.

### 5.2.4 Review the quality review checklists, and the deliverables submitted by the Design team

The SQA team will examine the Review Checklists depicted in the next section as well as the Designer Reports to determine any modifications that must be made to the design.

## 5.3 Document discrepancies using the attached review checklists.

The following checklists are provided for the project teams to report any design discrepancies that they have with the DFAS design specifications. These checklists show the are of review, and area to report the discrepancy. The Notes section at the bottom gives particular design specifications that should be reviewed as well as the Designer report that should be used to fill out the review.

## REVIEW FORM

<b>Business Unit &amp; Project</b>		<b>Review Date:</b>	<b>Author:</b>
<b>Version:</b> 1			
<b>Reviewers Names OR Associated Review Leader Form Reference:</b>		<b>Major / Minor:</b>	
<b>Outcome:</b> (Circle One) <b>ACCEPTED</b> (Once comments have been actioned) <b>NOT ACCEPTED</b> (Wish to re-review once comments			
<b>No</b>	<b>Action Items:</b>		
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
<b>Agreed by:</b>			
<b>Actions:</b>		<b>Proposed Completion Date:</b>	
<b>Follow-up Date:</b>		<b>Closure Signature &amp; Date:</b>	

\_\_\_\_\_

# REVIEW COMMENTS LIST

<b>Items Under Review:</b> (Table Notes to Revision History)		<b>Author:</b>
<b>Version:</b> 1		<b>Review Date:</b>
<b>Reviewers Names OR Associated Review Leader Form Reference:</b>		
<b>Outcome:</b> (Circle One) <b>ACCEPTED</b> (Once comments have been actioned) <b>NOT ACCEPTED</b> (Wish to re-review once comments		
<b>No</b>	<b>Table</b>	<b>Cat</b>
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
<b>Notes:</b> The table Notes should contain the revision history of the table. These include the date/time the table was originally created, and modifications made to the table <b>Standard:</b> 2.1 - 16 <b>Report:</b> Table Definitions Categories (Cat): MA - MAJOR MI - MINOR I - INFORMATION O - OBSERVATION Problem Types (Pt): M - MISSING W - WRONG E - EXTRA/SUPERFLUOUS NE - NEEDS EX CL - CLEARED (or tick)		

## REVIEW COMMENTS LIST

<b>Items Under Review:</b> (Naming Conventions)					<b>Author:</b>	
<b>Version:</b> 1			<b>Review Date:</b>			
<b>Reviewers Names OR Associated Review Leader Form Reference:</b>						
<b>Outcome:</b> (Circle One) <div style="display: flex; justify-content: space-between;"> <span><b>ACCEPTED</b> (Once comments have been actioned)</span> <span><b>NOT ACCEPTED</b> (Wish to re-review once comments)</span> </div>						
No	Category	Naming Convention	Cat	Pt	AC	
1	date/ month/ year/ time columns					
2	code/mnemonic/abbreviation columns					
3	amount columns					
4	currency columns					
5	change history columns					
6	description columns					
7	indicator columns					
8	status columns					
9	number columns					
10	sequence within parent columns					
<b>Notes:</b> The designers should have selected and used a consistent naming convention for each of the above categories. This review will li see if the conventions are used consistently.						
<b>Standard:</b> 2.2 - 9						
<div style="text-align: center;">             Categories (Cat):    MA - MAJOR    MI - MINOR    I - INFORMATION    O - OBSERVATIO              Problem Types (Pt):    M - MISSING    W - WRONG    E - EXTRA/SUPERFLUOUS    NE - NEEDS EX                 CL - CLEARED (or tick)           </div>						

# REVIEW COMMENTS LIST

Items Under Review: (Optional Columns Notes)					Author:
Version: 1			Review Date:		
Reviewers Names OR Associated Review Leader Form Reference:					
<b>Outcome:</b> (Circle One) <b>ACCEPTED</b> (Once comments have been actioned) <b>NOT ACCEPTED</b> (Wish to re-review once comments					
No	Table	Optional Column	Cat	Pt	AC
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
<b>Notes:</b> Columns that are optional should have a short column note that explains the meaning of a null value occurring for that column, i VALUE UNKNOWN. <b>Standard:</b> 2.2 - 22 <b>Report:</b> Table Definitions Categories (Cat): MA - MAJOR MI - MINOR I - INFORMATION O - OBSERVATION Problem Types (Pt): M - MISSING W - WRONG E - EXTRA/SUPERFLUOUS NE - NEEDS EX CL - CLEARED (or tick)					

## REVIEW COMMENTS LIST

<b>Items Under Review:</b> (Sequence Notes)						<b>Author:</b>	
<b>Version:</b> 1				<b>Review Date:</b>			
<b>Reviewers Names OR Associated Review Leader Form Reference:</b>							
<b>Outcome:</b> (Circle One) <div style="display: flex; justify-content: space-between;"> <span><b>ACCEPTED</b> (Once comments have been actioned)</span> <span><b>NOT ACCEPTED</b> (Wish to re-review once comments)</span> </div>							
No	Table	Sequence Definition	Cat	Pt	AC		
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
<b>Notes:</b> Sequences that are not ascending, incremented by 1, or cycle must be documented in the sequence description. <b>Standard:</b> 2.7 <b>Report:</b> Sequence Definition							
Categories (Cat): MA - MAJOR MI - MINOR I - INFORMATION O - OBSERVATION Problem Types (Pt): M - MISSING W - WRONG E - EXTRA/SUPERFLUOUS NE - NEEDS EXPLANATION CL - CLEARED (or tick)							

## REVIEW COMMENTS LIST

<b>Items Under Review:</b> (Table/Column Help Text recorded deviation from Analysis)  <b>Version:</b> 1	<b>Review Date:</b>	<b>Author:</b>			
<b>Reviewers Names OR Associated Review Leader Form Reference:</b>					
<b>Outcome:</b> (Circle One) <div style="display: flex; justify-content: space-around;"> <span><b>ACCEPTED</b> (Once comments have been actioned)</span> <span><b>NOT ACCEPTED</b> (Wish to re-review once comments)</span> </div>					
No	Table	Column	Cat	Pt	AC
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					

**Notes:** User/Help Texts are seen by the user as they navigate from one field to another on a screen. If the user/help text has been changed in the associated entity/attribute, the reason for the change must be documented in the table/column notes.

**Standard:** 2.1 - 17

**Report:** Table Definitions

Categories (Cat):

MA - MAJOR    MI - MINOR    I - INFORMATION    O - OBSERVATION

Problem Types (Pt):

M - MISSING    W - WRONG    E - EXTRA/SUPERFLUOUS    NE - NEEDS EXPLANATION

CL - CLEARED (or tick)

--

# REVIEW COMMENTS LIST

Items Under Review: (Table/Column Order)					Author:
Version: 1			Review Date:		
Reviewers Names OR Associated Review Leader Form Reference:					
<b>Outcome:</b> (Circle One) <b>ACCEPTED</b> (Once comments have been actioned) <b>NOT ACCEPTED</b> (Wish to re-review once comments					
No	Table	Column Order	Cat	Pt	AC
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
<b>Notes:</b> Columns must be defined in the following order: 1. primary key columns, 2. unique key columns, 3. foreign key columns, 4. all other optional columns. For clarity, it is sometimes better to keep columns together; for example: 'begindate' (mandatory) and 'enddate'					
<b>Standard:</b> 2.2 - 12					
<b>Report:</b> Table Definitions					
Categories (Cat): MA - MAJOR MI - MINOR I - INFORMATION O - OBSERVATION Problem Types (Pt): M - MISSING W - WRONG E - EXTRA/SUPERFLUOUS NE - NEEDS EX CL - CLEARED (or tick)					

## REVIEW COMMENTS LIST

Items Under Review: (Table Constraints)				Author:	
Version: 1				Review Date:	
Reviewers Names OR Associated Review Leader Form Reference:					
<b>Outcome:</b> (Circle One) <div style="display: flex; justify-content: space-between;"> <span><b>ACCEPTED</b> (Once comments have been actioned)</span> <span><b>NOT ACCEPTED</b> (Wish to re-review once comments)</span> </div>					
No	Table	Primary/Foreign/Unique/Check Constraints	Cat	Pt	AC
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
<b>Notes:</b> Examine how each constraint is to be implemented (e.g., Client, Server, or N-Tier). Cascading of foreign key updates and delete:					
<b>Standard:</b> 2.4					
<b>Report:</b> Tables, Columns and Foreign Key Derivations					
Categories (Cat): MA - MAJOR MI - MINOR I - INFORMATION O - OBSERVATION Problem Types (Pt): M - MISSING W - WRONG E - EXTRA/SUPERFLUOUS NE - NEEDS EXPLANATION CL - CLEARED (or tick)					

## REVIEW COMMENTS LIST

<b>Items Under Review:</b> (Module Definition)				<b>Author:</b>	
<b>Version:</b> 1				<b>Review Date:</b>	
<b>Reviewers Names OR Associated Review Leader Form Reference:</b>					
<b>Outcome:</b> (Circle One) <div style="display: flex; justify-content: space-around;"> <span><b>ACCEPTED</b> (Once comments have been actioned)</span> <span><b>NOT ACCEPTED</b> (Wish to re-review once comments)</span> </div>					
No	Module	Definition/Reason why not defined in 4GL	Cat	Pt	AC
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
<b>Notes:</b> Module names must be logical, without underscores and other special characters. If module is defined as using a 3GL, the reason must be stated in the comments section.					
<b>Standard:</b> 3.1.1					
<b>Report:</b> <div style="text-align: center; margin-top: 10px;">             Categories (Cat): MA - MAJOR MI - MINOR I - INFORMATION O - OBSERVATION              Problem Types (Pt): M - MISSING W - WRONG E - EXTRA/SUPERFLUOUS NE - NEEDS EXPLANATION              CL - CLEARED (or tick)           </div>					

## REVIEW COMMENTS LIST

<b>Items Under Review:</b> (Module User/Help Text)		<b>Author:</b>		
<b>Version:</b> 1		<b>Review Date:</b>		
<b>Reviewers Names OR Associated Review Leader Form Reference:</b>				
<b>Outcome:</b> (Circle One) <div style="display: flex; justify-content: space-around;"> <span><b>ACCEPTED</b> (Once comments have been actioned)</span> <span><b>NOT ACCEPTED</b> (Wish to re-review once comments)</span> </div>				
No	Module	Cat	Pt	
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				

**Notes:** The user help text must be targeted towards the future users of the application under development. It should at least contain a brief summarizing the functionality. The future users must be able to assess the usefulness and correctness of a module through the user help text validations performed by the module must be presented here, in a form understandable by the future users of the application for assessment

**Standard:** 3.1.2 - 5

**Report:**

Categories (Cat): MA - MAJOR MI - MINOR I - INFORMATION O - OBSERVATION

Problem Types (Pt): M - MISSING W - WRONG E - EXTRA/SUPERFLUOUS NE - NEEDS EXPLANATION

CL - CLEARED (or tick)

--

## REVIEW COMMENTS LIST

<b>Items Under Review:</b> (Module Post Generation Triggers)					<b>Author:</b>
<b>Version:</b> 1			<b>Review Date:</b>		
<b>Reviewers Names OR Associated Review Leader Form Reference:</b>					
<b>Outcome:</b> (Circle One) <div style="display: flex; justify-content: space-around;"> <span><b>ACCEPTED</b> (Once comments have been actioned)</span> <span><b>NOT ACCEPTED</b> (Wish to re-review once comments)</span> </div>					
No	Module	Post Generation Trigger	Cat	Pt	AC
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
<b>Notes:</b> The Notes of a screen module contain the code of any triggers and procedures that must be created or modified after generation. <b>Standard:</b> 3.1.2 - 4 <b>Report:</b> <div style="text-align: center; margin-top: 10px;"> Categories (Cat): MA - MAJOR MI - MINOR I - INFORMATION O - OBSERVATION  Problem Types (Pt): M - MISSING W - WRONG E - EXTRA/SUPERFLUOUS NE - NEEDS EXPLANATION  CL - CLEARED (or tick) </div>					

## REVIEW COMMENTS LIST

Items Under Review: (Module Parameters)					Author:
Version: 1			Review Date:		
Reviewers Names OR Associated Review Leader Form Reference:					
<b>Outcome:</b> (Circle One) <b>ACCEPTED</b> (Once comments have been actioned) <b>NOT ACCEPTED</b> (Wish to re-review once comments					
No	Module	Parameter Name	Cat	Pt	AC
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					

**Notes:** Name parameters using the convention, <<P>>\_<<logical\_name>>\_<<IN/OUT/INOUT>>. For parameters that are related to a name element of the parameter should be the column name. Always specify a sequence indicating the position of the parameter on the cc following datatypes for parameters: VARCHAR2, NUMBER, DATE. Parameter prompts should indicate what the parameter refers to. A adequately described

**Standard:** 3.1.6

**Report:** PL/SQL Module Definition

Categories (Cat):	MA - MAJOR	MI - MINOR	I - INFORMATION	O - OBSERVATION
Problem Types (Pt):	M - MISSING	W - WRONG	E - EXTRA/SUPERFLUOUS	NE - NEEDS EX
	CL - CLEARED (or tick)			

## REVIEW COMMENTS LIST

<b>Items Under Review:</b> (Module Table Details)					<b>Author:</b>	
<b>Version:</b> 1			<b>Review Date:</b>			
<b>Reviewers Names OR Associated Review Leader Form Reference:</b>						
<b>Outcome:</b> (Circle One) <div style="display: flex; justify-content: space-between;"> <span><b>ACCEPTED</b> (Once comments have been actioned)</span> <span><b>NOT ACCEPTED</b> (Wish to re-review once comments)</span> </div>						
No	Module	Tables	Cat	Pt	AC	
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
<b>Notes:</b> Define all tables and views used by this module, together with the way the tables and views are used (INSERT, UPDATE, DELETE, etc.) where/validation condition implements a documented business rule, provide a reference to this business rule. If the business rule is not yet defined, provide a description.						
<b>Standard:</b> 3.2.1						
<b>Report:</b> PL/SQL Module Definition						
Categories (Cat): MA - MAJOR MI - MINOR I - INFORMATION O - OBSERVATION Problem Types (Pt): M - MISSING W - WRONG E - EXTRA/SUPERFLUOUS NE - NEEDS EXPLANATION CL - CLEARED (or tick)						

## REVIEW COMMENTS LIST

<b>Items Under Review:</b> (Module Blocks)				<b>Author:</b>	
<b>Version:</b> 1				<b>Review Date:</b>	
<b>Reviewers Names OR Associated Review Leader Form Reference:</b>					
<b>Outcome:</b> (Circle One) <div style="display: flex; justify-content: space-around;"> <span><b>ACCEPTED</b> (Once comments have been actioned)</span> <span><b>NOT ACCEPTED</b> (Wish to re-review once comments)</span> </div>					
No	Module	Block Name	Cat	Pt	AC
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
<b>Notes:</b> Titles are singular only if the user commonly works with a single instance of data within the block; otherwise they are plural. If the block title is used as the title in the LOV window. Such titles should start with the verb Find, followed by the object name. <b>Standard:</b> 3.2 <b>Report:</b>					
Categories (Cat): MA - MAJOR MI - MINOR I - INFORMATION O - OBSERVATION Problem Types (Pt): M - MISSING W - WRONG E - EXTRA/SUPERFLUOUS NE - NEEDS EXPLANATION CL - CLEARED (or tick)					

--

## REVIEW COMMENTS LIST

<b>Items Under Review:</b> (Module Windows)					<b>Author:</b>
<b>Version:</b> 1			<b>Review Date:</b>		
<b>Reviewers Names OR Associated Review Leader Form Reference:</b>					
<b>Outcome:</b> (Circle One) <div style="display: flex; justify-content: space-around;"> <span><b>ACCEPTED</b> (Once comments have been actioned)</span> <span><b>NOT ACCEPTED</b> (Wish to re-review once comments)</span> </div>					
No	Table	Window Name	Cat	Pt	AC
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
<b>Notes:</b> Window titles are always plural, except when a user only works with a single instance of data. Secondary window titles display context can be made clear by merely showing primary key data, then context is indicated as <<standard window title>> (<<context>>), <					
<b>Standard:</b> 3.2.2					
<b>Report:</b>					
Categories (Cat): MA - MAJOR MI - MINOR I - INFORMATION O - OBSERVATION Problem Types (Pt): M - MISSING W - WRONG E - EXTRA/SUPERFLUOUS NE - NEEDS EXPLANATION CL - CLEARED (or tick)					

--

# REVIEW COMMENTS LIST

Items Under Review: (Module Columns)				Author:		
Version: 1			Review Date:			
Reviewers Names OR Associated Review Leader Form Reference:						
Outcome: (Circle One)						
ACCEPTED (Once comments have been actioned)			NOT ACCEPTED (Wish to re-review once comments			
No	Module	Column	Cat	Pt	AC	
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						

**Notes:** Columns should be defined using the following standards:

1. Do not deviate from the standard database column datatype for non-displayed columns.
2. Use a check box when only one value is applicable in a yes/no situation, and the yes/no statement is not contrived or obscure. A NULL, but it cannot be set to NULL.
3. Use a boolean set when a maximum of one of out of two values is applicable, and if the list will be static throughout the life of the set to NULL.
4. Use radio group, radio group (meaning) or radio group (abbreviation) when a maximum of one of two to five values is applicable throughout the life of the product.
5. Use pop list, pop list (meaning) or pop list (abbreviation) when only one of three to fifteen values is applicable, and the list is not fifteen.
6. Use LOV window when only one of five to twenty values is applicable, and the list is dynamic during the life of the product. (All tables should be displayed using a LOV window.
7. Avoid text list, text list (meaning) and text list (abbreviation) due to the amount of space they require. If using text lists, use the twenty entries.
8. Use combo box, combo box (meaning) or combo box (abbreviation) if you have a list of allowable values that will be used most often. The user knows that this list does not cover all situations, while at the same time the user is not able to complete the list.
9. Foreign key columns should be displayed in the same sequence as their primary key counterparts.
10. The first letter of any word in a prompt is capitalized. Prompts should clearly indicate to what property the column refers.
11. Hint texts take the form of the remainder of the sentence, "The value in this field registers <hint text>".
12. If you use the types Date Created, Date Modified, Created By and Modified By, the Forms Generator creates code in all related tables. However, the nature of such data auditing makes server-side implementation, using database triggers, imperative. Only use server-side implementation when these fields need to be displayed in the generated form. When using the autogenerate field type, use the preference WHTIME to include the time. Set the Display Datatype of the associated columns to Datetime.

**Standard:** 3.2.1

**Report:**

Categories (Cat):	MA - MAJOR	MI - MINOR	I - INFORMATION	O - OBSERVATION
Problem Types (Pt):	M - MISSING	W - WRONG	E - EXTRA/SUPERFLUOUS	NE - NEEDS EXPLANATION
		CL - CLEARED (or tick)		

**REVIEW COMMENTS LIST**

Items Under Review: (PL/SQL Blocks)				Author:		
Version: 1			Review Date:			
Reviewers Names OR Associated Review Leader Form Reference:						
Outcome: (Circle One)						
ACCEPTED (Once comments have been actioned)			NOT ACCEPTED (Wish to re-review once comments			
No	Module	PL/SQL Block	Cat	Pt	AC	
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						

**Notes:** Enter entire PL/SQL block, include a header with items such as the following: purpose, parameters, revision history

The following reviews should be performed on PL/SQL code modules:

1. PL/SQL code should be divided into 2 segments:
  - Declarative part containing variable declarations, implicit conversions, cursor declarations, and exception declarations
  - Executable part starting with BEGIN, containing exception code starting with EXCEPTION , and ending with END
2. Local variable should be declared in the Declarative part and not have the same name as table columns or be reserved words
3. Explicitly code all transaction control yourself.
4. Do not interfere with the Oracle Forms transaction mechanisms from within a PL/SQL code segment using COMMIT, SAVEPOINT, or ROLLBACK. The only exception is POST, which you are allowed to use in combination with the Oracle Server with the transaction processing
5. In Oracle Forms, only use anonymous blocks if a PL/SQL block consists of only a single expression; for example, the call of
6. Avoid replicating code. Make use of stored procedures and functions or libraries as much as possible.
7. Use “C” style comments /\* \*/ instead of –
8. Use the following standards for declarations:
  - Begin cursors with c\_
  - When converting datatypes using a formatting function, add the following suffix to target datatypes: “\_c” for character, number
  - Name temporary storage of columns with the prefix “t\_” and the column name
  - Name exceptions starting with “e\_”
  - Declare variables that will hold column information as type “%type”
  - Declare variables that will hold row data as type “%rowtype”
9. If the designers use the raise\_application\_error exception handlers, use errors numbers -20000 through -20999 (Other numbers are reserved for Oracle)
10. Customize exception handlers message text explicitly to pass module-relevant messages.
11. Avoid implicit datatype conversion. If you expect implicit datatype conversion to occur, you must place an explicit datatype conversion expression.
12. Make sure that in each of the loop constructs the condition to end the execution of the loop will eventually occur. Be careful with values in the loop condition that may cause early loop termination or no loop termination at all.
13. Use a cursor FOR loop as the preferred method of handling SELECT statements that return more than one row.
14. Do not place excessive coding in the exception-handling part of a PL/SQL block; only specify the code that is strictly necessary to handle the exception effectively.

**Standard:** 3.1.2

**Report:** PL/SQL Module Definition

Categories (Cat):	MA - MAJOR	MI - MINOR	I - INFORMATION	O - OBSERVATION
Problem Types (Pt):	M - MISSING	W - WRONG	E - EXTRA/SUPERFLUOUS	NE - NEEDS EXPLANATION
	CL - CLEARED	(or tick)		

## REVIEW COMMENTS LIST

Items Under Review: (Storage Volumes)					Author:
Version: 1			Review Date:		
Reviewers Names OR Associated Review Leader Form Reference:					
Outcome: (Circle One)					
ACCEPTED (Once comments have been actioned)			NOT ACCEPTED (Wish to re-review once comments)		
No	Database/Tablespace/User/Table/ Index	Initial Size/End Size/Max Size	Cat	Pt	AC
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
<b>Notes:</b> The size calculations are derived from the number of rows that will be stored in each base table, and is calculated in the Database Size Estimates Report provided as part of Oracle Designer. To run the report, the table, and keys must be assigned to a Database, Tablespace, and Index. <b>Standard:</b> 2.1 - 8 <b>Report:</b> Database Table and Index Size Estimates					
Categories (Cat): MA - MAJOR MI - MINOR I - INFORMATION O - OBSERVATION Problem Types (Pt): M - MISSING W - WRONG E - EXTRA/SUPERFLUOUS NE - NEEDS EXPLANATION CL - CLEARED (or tick)					





REVIEW COMMENTS LIST

Items Under Review: (Tables to Entity Mapping)					Author:
Version: 1			Review Date:		
Reviewers Names OR Associated Review Leader Form Reference:					
Outcome: (Circle One)					
ACCEPTED (Once comments have been actioned)			NOT ACCEPTED (Wish to re-review once comments		
No	Table/Column	Entity/Attribute	Cat	Pt	AC
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
<b>Notes:</b> Requirements Traceability is critical to the success of DFAS. Each table/column that is not a primary or foreign key must be directly linked to an entity/attribute. Any tables/columns which cannot be traced to an entity/attribute, and the reason for their creation including why not entity/attribute documented in the notes for that table/column.					
<b>Standard:</b> 1.2 - 6					
<b>Report:</b> Entity to Table Implementation					
Categories (Cat): MA - MAJOR MI - MINOR I - INFORMATION O - OBSERVATION					
Problem Types (Pt): M - MISSING W - WRONG E - EXTRA/SUPERFLUOUS NE - NEEDS EXPLANATION					
CL - CLEARED (or tick)					

--

## REVIEW COMMENTS LIST

Items Under Review: (Module to Function Mapping)						Author:	
Version: 1				Review Date:			
Reviewers Names OR Associated Review Leader Form Reference:							
<b>Outcome:</b> (Circle One) <div style="display: flex; justify-content: space-around;"> <span><b>ACCEPTED</b> (Once comments have been actioned)</span> <span><b>NOT ACCEPTED</b> (Wish to re-review once comments)</span> </div>							
No	Module	Function	Cat	Pt	AC		
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
<b>Notes:</b> Requirements Traceability is critical to the success of DFAS. Each module must be associated with one or more functions. Any module must have at least one function, and the reason for their existence and why no function exists must be documented in the module.							
<b>Report:</b> PL/SQL Module Definition, Module Definition							
<div style="text-align: center;">             Categories (Cat): MA - MAJOR MI - MINOR I - INFORMATION O - OBSERVATION              Problem Types (Pt): M - MISSING W - WRONG E - EXTRA/SUPERFLUOUS NE - NEEDS EXPLANATION              CL - CLEARED (or tick)           </div>							

--

**DCII CRITICAL DESIGN REVIEW CHECKLIST (CDR)**  
**FOR \_\_\_\_\_**  
**(Applications)**

Mark each question (Y) for Yes, (N) for No , (N/A) for Non-applicable, or (NR) for Not Reviewed.

<b>Project Management Deliverables</b>	<b>Accepted? Y-N-N/A-NR</b>	<b><u>COMMENTS</u></b>
Logical Database Design		
Module Functional Documentation		
Module Technical Documentation		
Menu Structure		
Audit Facilities		

**Designer Reports**

The following reports are organized according to the Project Management Deliverables. The Project Management Deliverables correspond to having a complete set of the Designer Deliverables.

Mark each question (Y) for Yes, (N) for No, (N/A) for Non-applicable, or (NR) for Not Reviewed.

<b>Designer Reports Deliverables</b>	<b>Accepted? Y-N-N/A-NR</b>	<b><u>COMMENTS</u></b>
--------------------------------------	---------------------------------	------------------------

**Logical Database Design**

Table Definition		
Snapshot/View Definition		
Column Definition		
Constraint Definitions		
Tables and Primary Key Derivations		
Database Trigger		
Database Synonym Definition		
Columns in a Domain		

### Module Functional Documentation

Module Definition		
Module Network		
Module Documentation		

### Module Technical Documentation

Module Program Data Usages		
Module Argument		
PLSQL Module Definition		
Detailed Module Definition		
Module Program Data		

### Menu Structure

Menu and Screen Definition		
Module Component Definition		
Module Network		

### Audit Facilities

Column Change Impact Analysis		
Column Display Usage (by table)		

### OVERALL RATINGS

Requirement	Y/N	Comments
Is the design complete?		
Is the design accurate?		
Is design sufficient for development to begin?		

#### 5.4. Approval / disapproval to proceed

Once the SQA team has examined the design reviews checklists, the team will decide whether the project is ready to go on to the next step, which is to participate in the ICDR. At that time, the SQA team will approve the design. The DFAS project team will use the following letter to indicate the approval of the design and the readiness to proceed with the build phase:

##### **STATEMENT OF AGREEMENT**

For the Program Manager (Bruce Johnson): My signature below signifies that, once any outstanding Action Items identified during the Functional Requirements Review on \_\_\_\_\_, 1999 are completed, the requirements for DFAS Corporate Database Release 9902 are complete and accurate.

For the Technical Project Officer (Gini Calchera): My signature below signifies that, once any outstanding Action Items identified during the Functional Requirements Review on \_\_\_\_\_, 1999 are completed, the requirements for DFAS Corporate Database Release 9902 are sufficiently understood for system design and development to proceed.

\_\_\_\_\_  
Bruce Johnson  
DCD Program Manager

\_\_\_\_\_  
Gini Calchera  
DCD Technical Project Officer

If however, the SQA team feels that the number and type of discrepancies would require that the project team make changes to the design, the SQA team will disapprove the design and provide to the project team a list of all modifications that must be made to the design to prepare for the ICDR. The following letter will be used by the DFAS team to disapprove the design and indicate that further work is necessary before the application is ready to proceed to build.

## STATEMENT OF DEFICIENCY

For the Program Manager (Greg Williams): My signature below signifies that the requirements reviewed at the DFAS Corporate Database Functional Requirements Review on June 10, 1999 are not complete and accurate.

For the Technical Project Officer (Gini Calchera): My signature below signifies that the requirements reviewed at the DFAS Corporate Database Functional Requirements Review on June 10, 1999 are not sufficiently understood for system design and development to proceed.

\_\_\_\_\_  
Greg Williams  
DCD Program Manager

\_\_\_\_\_  
Gini Calchera  
DCD Technical Project Officer

If the DCII SQA has been monitoring the integrated Design products on an ongoing basis, the formal ICDR may be short.

### 5.5 Follow-up

The SQA team will maintain contact with project team after review to assist the project team in making suggested changes to application design. The SQA team will also work with project team to determine any changes that require generation of SCR

The project team will then schedule a follow-up review when they have completed the recommended changes.

### ***Complete Review Defect Report***

#### Purpose

The software development team will provide a system overview of the SCRs under consideration. The detailed design is then presented followed by discussions, questions and concerns. Design is validated against SCRs for accuracy, clarity, completeness, consistency, testability, and feasibility. The CDR checklists are then annotated and the following items must be documented for each product defect noted. This information will be included in the CDR Summary Report.

1. Identification of the release and SCR
2. Product being reviewed
3. Description of product defect
4. Origin of product defect (e.g., requirements definition, system specification, design specification, program name, etc.)
5. Determine category and severity of defect (Categories: M = missing, E = extra, W = wrong) (Severity: Major = will prevent user from getting work done, Minor = noticeable, but doesn't interfere with work)
6. Corrective actions required for defect (if known)
7. Action item assignee (if known)
8. Person responsible for defect correction (if known)

An automated log should be maintained of all the information resulting from items mentioned above.

## ***Complete Review Report***

### **Purpose**

The CDR Summary Report will be prepared by the review coordinator and distributed to CDR participants and appropriate management/project personnel as identified in the SQA Plan.

Description: Prepare the CDR Summary Report to include review results and recommendations for corrections.

1. Date and time review took place
2. System/Project Identification
3. CDR Participants/Organizational Element
4. List of SCRs Reviewed
5. Review results
6. Information collected in the CDR Validation procedure
7. Recommendations/Action Items

Description: Distribute the CDR Summary Report to participants and appropriate management/project personnel as identified in the SQA Plan, ensuring a copy is available to the Staff SQA.

## ***Complete Review Checklist***

### **Purpose**

Description: Using the SQA Guidelines, tailor/expand the CDR checklist, if necessary, for the SCRs under consideration. Include this checklist in the review package for the participants.

## ***Forward Checklist to SQA***

### **Purpose**

Description: SQA will review selected design products for compliance with product and software development standards as identified in the SQA Plan. Any areas of non-compliance are documented for inclusion in the CDR Summary Report. Any request for waiver must have been requested on or before the date of the review. SQA will also ensure the checklist/ questionnaires for each review are completed and recommend any changes where items/questionnaires appear to be inappropriate. If SQA is unable to attend the CDR, the review coordinator will assign the SQA compliance role to one of the other participants.

## ***Manage Risks***

### **Purpose**

The Oracle Designer system is oriented to developing applications by managing the necessary life-cycle documentation, and using generally accepted standards to convert this documentation to working applications. It has been developed to create applications which are efficient and are well integrated with the Oracle database, forms, and web. The transformers and generators that are included in Designer will not create baseline applications unless the life-cycle documentation is sufficiently detailed.

A major focus of Designer is to make sure that all system components can be traced back to the documented requirements. Herein lies the predominant risks that must be managed. Namely including designs objects that have not been created using the transformer.

Specifically the following risks exist:

1. Defining/using tables that have not been transformed from entities
2. Defining/using columns that have not been transformed from attributes
3. Defining modules that have not been transformed from functions

To minimize the risks, the designers need to adequately document why any of these objects have been designed which have not been transformed from the requirements.

A second area of risk is to make modifications to design level objects in ways that do not follow the DFAS Design Standards. These standards have been developed to make sure that design objects can be generated using the Designer application generator.

## ***Record date accepted by Development***

### **Purpose**

Once the CDR has been completed and all reviewers are satisfied that the Design Model is properly documented, the date that the reviews are completed must be recorded in the project schedule.

## ***Update Technical Design Schedule***

### **Purpose**

The completion of the Design Model is a critical milestone in the Technical Design Schedule for the application. Since this date might not match the planned date, the Technical Design Schedule must be updated to match the Design completion date, and the future milestones will have to be rescheduled.

## **6. Verification:**

- a. SQA Review of Process
- b. SQA Audit of Product

## 7. Exit Criteria:

- a. Updated Technical Design Schedule (S-PM-011)
- b. Documented Risks (S-PM-013)
- c. Completed Critical Design Review Report (S-SE-001)
- d. Completed Critical Design Review Defect Report (S-SE-002)
- e. Completed Critical Design Review Checklist (S-PM-018)

## 8. Measures:

Data Collected for each Review

Type of Review

Date of Review

Number of SCRs reviewed

Duration of Review (In Hours)

Number of participants

Number of Saves by Origin

Number of Saves by Cause

Number of Saves by Priority

Numerical Value of Checklist

Data Collected for each Defect

Effort Required to Resolve Defect

Data Collected for each SCR

Revised Stop Date

Revised Size of Change

Revised Effort

Data Collected for each Risk

Priority

Date Identified

Status

Date Closed

Data Collected for each Action Item Generated

Responsibility

Resolution Date